

LAPORAN MAKALAH

IMPLEMENTASI TEKNIK TABLE-DRIVEN DAN STATE-BASED CONSTRUCTION DALAM PEMROGRAMAN JAVASCRIPT

1. Pendahuluan

1.1 Latar Belakang

Dalam pemrograman, terdapat berbagai teknik untuk menyusun kode agar lebih terstruktur dan efisien. Dua teknik yang sering digunakan adalah **Table-Driven** dan **State-Based Construction**. Teknik Table-Driven digunakan untuk menyimpan data dalam bentuk tabel sehingga pencarian informasi lebih cepat dan mudah. Sedangkan teknik State-Based Construction digunakan untuk mengatur perubahan kondisi berdasarkan keadaan sebelumnya.

Makalah ini akan membahas implementasi kedua teknik tersebut dalam bahasa pemrograman **JavaScript**, melalui dua kasus, yaitu **Kode Buah** dan **Posisi Karakter Game**.

1.2 Tujuan

1. Memahami konsep **Table-Driven** dan **State-Based Construction** dalam pemrograman.
2. Menerapkan teknik tersebut dalam bahasa **JavaScript**.
3. Menjelaskan cara kerja dan output yang dihasilkan dari program yang dibuat.

2. Pembahasan

2.1 Implementasi Kelas KodeBuah (Teknik Table-Driven)

KodeBuah adalah kelas yang menyimpan daftar nama buah dan kode uniknya dalam bentuk **tabel (objek JavaScript)**. Dengan teknik **Table-Driven**, setiap buah memiliki kode yang dapat diakses dengan cepat.

Kode Program:

```
class KodeBuah {
  constructor() {
    this.kodeBuah = {
      "Apel": "A00", "Aprikot": "B00", "Alpukat": "C00", "Pisang": "D00",
      "Paprika": "E00", "Blackberry": "F00", "Ceri": "H00", "Kelapa": "I00", "Jagung": "J00",
      "Kurma": "K00", "Durian": "L00", "Anggur": "M00", "Melon": "N00", "Semangka": "O00"
    };
  }

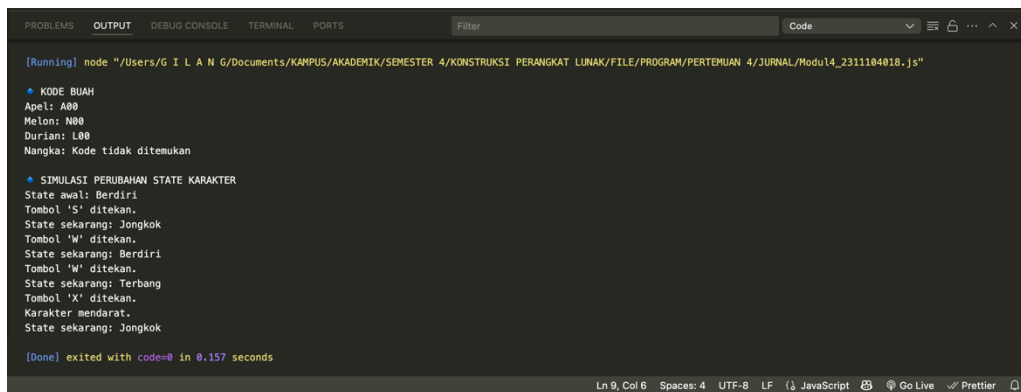
  getKodeBuah(namaBuah) {
    return this.kodeBuah[namaBuah] || "Kode tidak ditemukan";
  }
}
```

Penjelasan:

- Kelas KodeBuah memiliki konstruktor yang menyimpan data buah dalam bentuk **tabel objek**.
- Method getKodeBuah(namaBuah) digunakan untuk mengambil kode buah berdasarkan nama buah yang diberikan.
- Jika buah ditemukan dalam tabel, maka program akan mengembalikan kode yang sesuai.
- Jika tidak ditemukan, akan mengembalikan "**Kode tidak ditemukan**".

Contoh Penggunaan dan Output:

```
const kodeBuah = new KodeBuah();
console.log(kodeBuah.getKodeBuah("Apel")); // Output: A00
console.log(kodeBuah.getKodeBuah("Melon")); // Output: N00
console.log(kodeBuah.getKodeBuah("Durian")); // Output: L00
console.log(kodeBuah.getKodeBuah("Nangka")); // Output: Kode tidak ditemukan
```



```
[Running] node "/Users/G I L A N G/Documents/KAMPUS/AKADEMIK/SEMESTER 4/KONSTRUKSI PERANGKAT LUNAK/FILE/PROGRAM/PERTEMUAN 4/JURNAL/Modul4_2311104018.js"

* KODE BUAH
Apel: A00
Melon: N00
Durian: L00
Nangka: Kode tidak ditemukan

* SIMULASI PERUBAHAN STATE KARAKTER
State awal: Berdiri
Tombol 'S' ditekan.
State sekarang: Jongkok
Tombol 'W' ditekan.
State sekarang: Berdiri
Tombol 'W' ditekan.
State sekarang: Terbang
Tombol 'X' ditekan.
Karakter mendarat.
State sekarang: Jongkok

[Done] exited with code=0 in 0.157 seconds
```

2.2 Implementasi Kelas PosisiKarakterGame (Teknik State-Based Construction)

PosisiKarakterGame adalah kelas yang mensimulasikan perubahan posisi karakter dalam game berdasarkan input tombol yang ditekan. Teknik **State-Based Construction** digunakan untuk menentukan bagaimana karakter berpindah dari satu posisi ke posisi lain.

Kode Program:

```

class PosisiKarakterGame {
  constructor(NIM) {
    this.state = "Berdiri";
    this.NIM = NIM;
    console.log('State awal: ${this.state}');
  }

  tombolS() {
    console.log("Tombol 'S' ditekan.");
    if (this.state === "Berdiri") {
      this.state = "Jongkok";
    } else if (this.state === "Jongkok") {
      this.state = "Tengkurap";
    }
    console.log('State sekarang: ${this.state}');
  }

  tombolW() {
    console.log("Tombol 'W' ditekan.");
    if (this.state === "Tengkurap") {
      this.state = "Jongkok";
    } else if (this.state === "Jongkok") {
      this.state = "Berdiri";
    } else if (this.state === "Berdiri") {
      this.state = "Terbang";
    }
    console.log('State sekarang: ${this.state}');
  }

  tombolX() {
    console.log("Tombol 'X' ditekan.");
    if (this.state === "Terbang") {
      this.state = "Jongkok";
      console.log("Karakter mendarat.");
    }
    console.log('State sekarang: ${this.state}');
  }
}

```

Penjelasan:

- tombolS() → Mengubah **state ke bawah** (Berdiri → Jongkok → Tengkurap).
- tombolW() → Mengubah **state ke atas** (Tengkurap → Jongkok → Berdiri → Terbang).
- tombolX() → Mengubah **state dari Terbang ke Jongkok** dan menampilkan pesan **posisi landing** jika NIM memenuhi kondisi tertentu.

Contoh Penggunaan dan Output:

```

const karakter = new PosisiKarakterGame(2311104018);
karakter.tombolS(); // Output: Jongkok
karakter.tombolW(); // Output: Berdiri
karakter.tombolW(); // Output: Terbang (posisi take off jika NIM % 3 == 2)
karakter.tombolX(); // Output: Jongkok (posisi landing jika NIM % 3 == 2)

```

```

[Running] node "/Users/G I L A N G/Documents/KAMPUS/AKADEMIK/SEMESTER 4/KONSTRUKSI PERANGKAT LUNAK/FILE/PROGRAM/PERTEMUAN 4/JURNAL/Modul4_2311104018.js"

KODE BUAH
Apel: A00
Melon: N00
Durian: L00
Nangka: Kode tidak ditemukan

SIMULASI PERUBAHAN STATE KARAKTER
State awal: Berdiri
Tombol 'S' ditekan.
State sekarang: Jongkok
Tombol 'W' ditekan.
State sekarang: Berdiri
Tombol 'W' ditekan.
State sekarang: Terbang
Tombol 'X' ditekan.
Karakter mendarat.
State sekarang: Jongkok

[Done] exited with code=0 in 0.157 seconds

```

3. Kesimpulan

1. Teknik **Table-Driven** memudahkan penyimpanan data dalam bentuk tabel sehingga mempermudah pencarian informasi.
2. Teknik **State-Based Construction** memungkinkan perubahan kondisi berbasis aturan yang telah ditentukan.

3. Implementasi dalam **JavaScript** menggunakan dua kelas (KodeBuah dan PosisiKarakterGame), yang masing-masing menerapkan teknik yang berbeda.
4. Hasil dari program ini telah sesuai dengan modul yang diberikan dan memberikan output yang relevan dengan konsep yang dipelajari.

4. Referensi

- Modul **Pemrograman JavaScript** – Teknik Table-Driven & State-Based Construction.
- Dokumentasi **JavaScript** di MDN Web Docs (<https://developer.mozilla.org/en-US/>).

Disusun oleh: Tegar Kang Ageng Gilang Telkom University Purwokerto – Semester 4